



Escuela Técnica Superior de Ingeniería del Diseño



**REMOTE CONTROL OF A MECHATRONIC SYSTEM WITH
WIRELESS COMMUNICATIONS**

FINAL MASTER THESIS

MASTERS IN MECHATRONICS ENGINEERING

AUTHOR: BHARATH KUMAR MITTAPALLI

TUTOR : D. JUAN JOSÉ SERRANO MARTÍN

DEPARTAMENTO DE INFORMÁTICA DE SISTEMAS Y COMPUTADORES

SEPTEMBER 2016



Contents

1.	INTRODUCTION.....	5
1.1	PURPOSE OF THE PROJECT.....	5
1.1.1	PROJECT INTEREST	5
1.1.2	OBJECT OF STUDY	5
1.1.3	ASPECTS INCLUDED.....	5
1.2	FEASIBILITY OF THE PROJECT	6
1.2.1	ASSESSMENT OF AVAILABLE RESOURCES	6
1.2.2	ORIENTATION OF THE PROJECT	6
1.3	BACKGROUND AND MOTIVATION	7
1.4	OBJECTIVES	7
1.5	STUDY PROCESS	8
2.	ORINATION	10
2.1	INTODUCTION TO ROBOTICS	10
2.2	DESIGN OF INDUSTRIAL ROBOT.....	13
2.3	APPLICATIONS OF INDUSTRIAL ROBOTS.....	14
2.4	BACKGROUND OF CNC.....	14
2.5	EVOLUTION OF CNC	16
3.	DESIGN OF CNC SYSTEM	18
3.1	CONSTRUCTION OF THE SYSTEM.....	18
3.2	DC MOTORS	18
3.3	DRIVER CIRCUIT OF MOTOR.....	24
3.4	LIMIT SWITCHES.....	25
3.5	MICROCONTROLLER.....	27
3.6	SOFTWARE	27
3.7	CIRCUIT CONNECTIONS.....	28
4.	STM32 F303VCT6 MICROCONTROLLER	32
4.1	INTRODUCTION TO MICROCONTROLLERS.....	32
4.2	STM32 F3 DISCOVERY BOARD.....	33



4.3	STM32 CUBEMX	35
5.	RF COMMUNICATION MODULE.....	38
5.1	INTRODUCTION	38
5.2	RF MODULE	38
5.3	TELIT LE50-868	39
5.4	CONFIGURATION OF TELIT LE50-868	41
5.5	ANALOG JOYSTICK.....	44
6.	SOFTWARE PROGRAMMING.....	47
6.1	KEIL SOFTWARE.....	47
6.2	BLOCK DIAGRAM.....	49
6.3	PROGRAM	52
7.	RESULT AND CONCLUSION	57
7.1	FUTURE SCOPE	57
8.	REFERENCES.....	60
8.1	BOOKS	60
8.2	MANUALS.....	60
8.3	WEB.....	60

List of Figures

Figure 1. Types of Robots.....	12
Figure 2. Configuration of Industrial Robots.....	13
Figure 3. Fleming's Left Hand Rule	19
Figure 4. Block diagram of DC motor	20
Figure 5. Types of DC motors	20
Figure 6. Kelvin K30 37.62 DC Motor.....	23
Figure 7. L298N Driver of ST Electronics	24
Figure 8. Limit Switch	25
Figure 9. Working of a Limit Switch.....	26
Figure 10. Schematic of the system	27
Figure 11. Circuit of DC motors connection to microcontroller	28
Figure 12. Circuit of limit switches connections to microcontroller	29
Figure 13. The project machine after all the connections	30
Figure 14. Schematic of STM32 F3 Discovery Board	33
Figure 15. STM32CubeMX main window after configuring the microcontroller.	36
Figure 16. Telit LE50-868 RF Module with its support board.	40
Figure 17. S-One management tool opening window	41
Figure 18. The Configuration Wizard window of S-One management tool.....	42
Figure 19. Analog Joystick	44
Figure 20. Block Diagram of the Mechatronic system of the project.....	50
Figure 21. General flow chart of work flow	51



Escuela Técnica Superior de Ingeniería del Diseño



CHAPTER 1

INTRODUCTION



1. INTRODUCTION

1.1 PURPOSE OF THE PROJECT

1.1.1 PROJECT INTEREST

This project aims to the study, design and implementation of an embedded system and for the re-modeling of a Mechatronic machine in use. This machine is from the department of Informatics systems and Computers of UPV. The main work in this project is to upgrade the hardware and software configuration with alternate components which we have studied in the subject of "Embedded systems".

By doing so, this makes it more easy to use in the practical works of the class of master of mechatronics. And also the design becomes more economical. We are also going include a radio frequency communication module with which we can control the whole system in wireless mode.

1.1.2 OBJECT OF STUDY

The machine that we are discussing here is a similar model of a Computer Numeric Control (CNC) machine. These types of machines are on a high demand these days all over the world. 3D printing is one of the major applications of such machine. Our main aim is to make this machine be used for different applications and in a simpler way.

The machine in use here is run by the DC motors. The hardware present in the machine is replaced by Microcontroller-based devices which we have studied in the class. These devices will help control the movement of the motors on the work surface.

1.1.3 ASPECTS INCLUDED

The constructed model of the machine is already available in the laboratory. So we just study and analyze about the design of the machine. We only make the necessary changes of the hardware components which makes the functioning of the machine in a simpler way that we



are expecting for. Changes in the design are also made if necessary. The software program is written for the added micro-controller devices.

1.2 FEASIBILITY OF THE PROJECT

1.2.1 ASSESSMENT OF AVAILABLE RESOURCES

Based on the material available in the laboratory of Informatics systems and Computers we have all the components and tools required for us.

We have the constructed structure of the machine, cables, tools, DC motors, the electronic components required for the upgrade of the machine. We can use the same model of machine for different applications since the programming and functioning will almost be the same or just has some slight changes.

The software programming for the micro-controller is written using the suite “Keil uvision5” which we have learned in the master course. Other management tools “Cube” for the micro-controller settings and “Telit” for the radio frequency module settings are used at some stages.

1.2.2 ORIENTATION OF THE PROJECT

The work of this project is done in the ITACA laboratory of the department of Informatics systems and Computers in UPV during the opening hours. Our goal is to control the movement of motors and also in the wireless mode. We upgrade it in such a way and make it available for teaching and for the use of lab work of the master course.



1.3 BACKGROUND AND MOTIVATION

Today the robotic technology is advancing very quickly. Constant study, development of new technologies and processes require highly accurate and fast manufacturing, where machine tools play a key role. The CNC machines are used in industry for years, but they have been modernized in recent years.

Thus we find the need to understand the operation of these machines in order to design them and then sell them. In this thesis we discuss the movement of the CNC machine in an efficient way. We analyze the construction of the machine and upgrade it as per our requirement and work on it to make it function in a simpler way. For this we study all about the electronic components, drivers used for the control of motors, mechanical functioning and software programming of the machine which combined together gives us required result.

1.4 OBJECTIVES

The objective of the work is to build a prototype of an industrial CNC machine of X-Y axes and make it functioning with best possible performance. Our main aim is to improve the characteristics of the machine and upgrade it for an efficient usage and achieve speed, accuracy and stability.

Specific goals:

- Develop an embedded system for the movement of the CNC machine.
- Modify the machine as per our requirements with micro-controller devices and end stop push buttons.
- Include a radio frequency communication module to control the whole machine in wireless mode.
- Check the operation of the prototype.



1.5 STUDY PROCESS

Work for the development of the problem is divided into different stages:

- Specifications: The machine will be able to play a 2D CNC model that will be processed by software and performed by the microcontroller.
- Study of the components to be used: The choice of suitable electronics, the type of motor control, the micro-controller device and connection system is studied.
- Program Design in C (Keil): A study of the software is done and programming will be done according to this.
- Choice of Control: For the control of machine we introduce a radio frequency communication module. Then we configure this module and use this for the control of the machine.



Escuela Técnica Superior de Ingeniería del Diseño



CHAPTER 2

ORINATION



2. ORIGINATION

2.1 INTRODUCTION TO ROBOTICS

A robot is defined as a reprogrammable, multifunctional manipulator designed to move material, parts, tools or specialized devices through variable programmed motions for the performance of a variety of tasks: A per Robot Institute of America, 1979.

In a simpler way a robot can also be defined as a machine which is designed and programmed to do the work of a human in any field or industry. The robots are mostly used for the jobs which are too dangerous for a human to do and also for the repetitive jobs that are boring, stressing and which includes a lot of man power. The study related to the design, programming and implementation of Robots is known as Robotics.

Day by day the technology used in the field of robotics kept on increasing with the introduction of microprocessors, micro-controllers, advanced sensors, and many other control devices. Since past few decades we have seen an exponential growth in robotics and we are implementing it in almost all the fields and for industrial processes. With more technological advancements of electronics, mechanics and computing systems in this field has all put together to be called as “Mechatronics systems” these days.

We can categorize the robots into different generations depending upon their development since the past few decades as:

First Generation Robots: These robots are programmed to perform a repeated sequence of tasks. They are unable to generate a feedback and cannot change their functionality according to the changes in the surroundings.



Second Generation Robots: These generation robots are capable of sensing the little changes in the surroundings and function accordingly by modifying the way they work. They are able to generate a feedback.

Third Generation Robots: The third generation of robots employs the concept of artificial intelligence. They make use of advanced microprocessors and control systems and are capable of logical reasoning according to the changes that occur in the surroundings.

Fourth Generation Robots: The concept of this type of robots is to reproduce and evolve themselves in number and performance. Nowadays people are working seriously on this idea to achieve success in this. They are not yet into existence.

- Some of the pictures of Robots are given below:

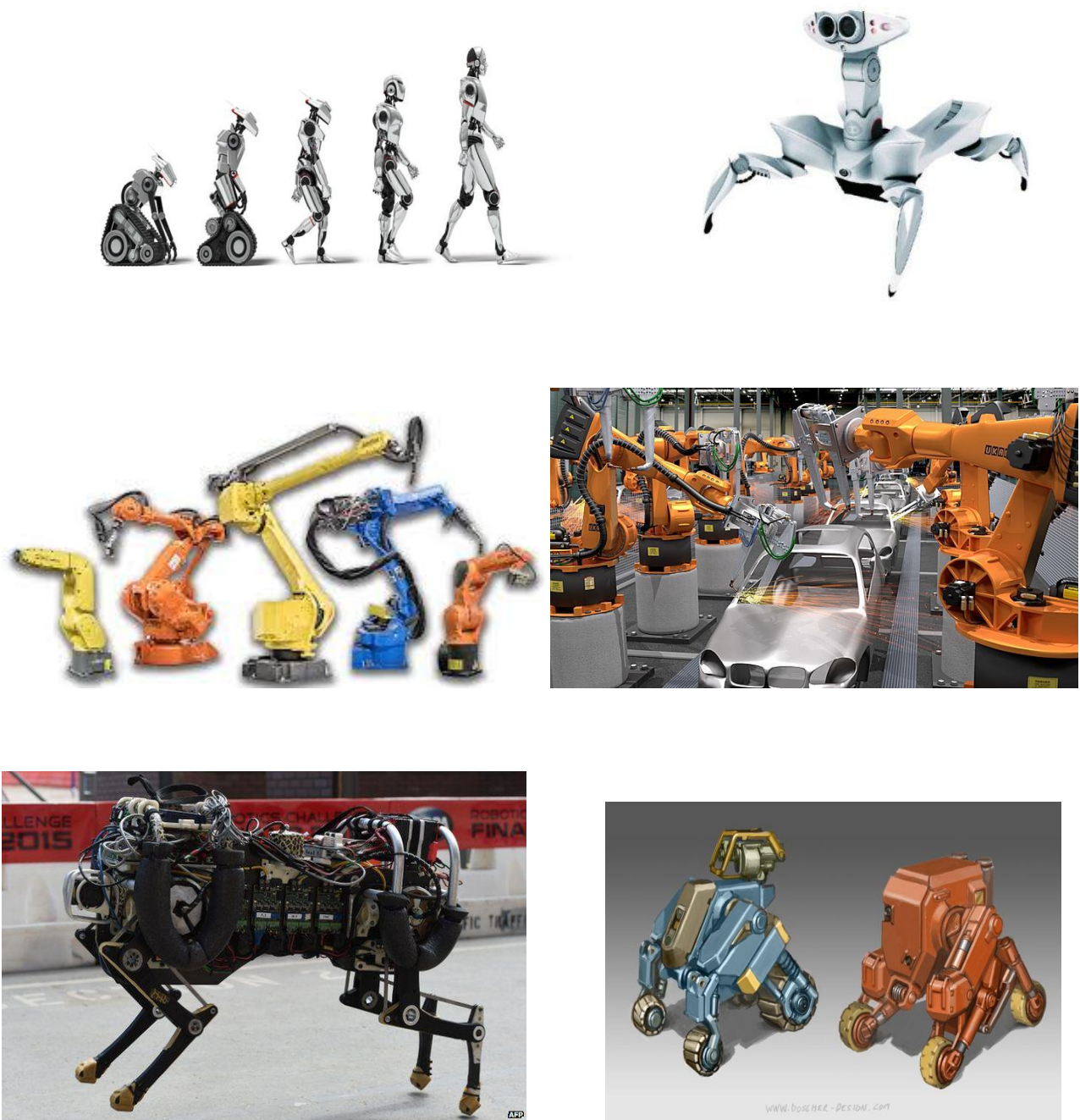


Figure 1. Types of Robots

2.2 DESIGN OF INDUSTRIAL ROBOT

An industrial robot consists of a mechanical structure with motors, gearboxes, sensors, actuators, drivers, and control system and end terminals. All these components are designed and fitted in a best architecture to make the robot function in an efficient way.

Mechanically, an industrial robot is composed of elements or links connected by joints that allow relative movement. Each joint can make linear, angular displacements or a combination of both.

The independent movement of each single joint is called its degree of freedom. And the degree of freedom of a robot is given by the sum of the degrees of freedom of all the joints that compose it.

The use of different combination of joints gives rise to different configuration of robots, which depends entirely on the applicability of the robot. In this project we use a Cartesian robot which has a linear motion in X-Y axes.

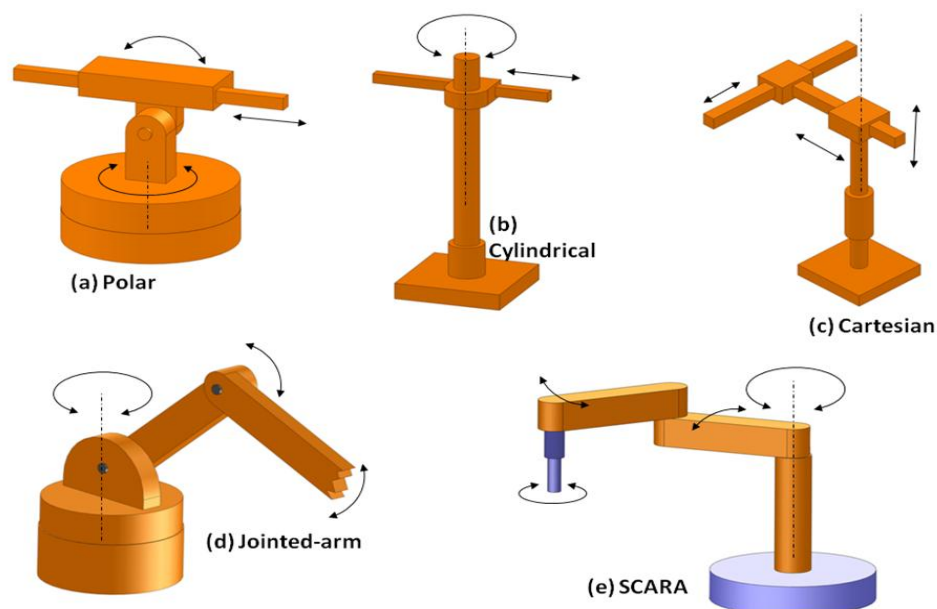


Figure 2. Configuration of Industrial Robots



2.3 APPLICATIONS OF INDUSTRIAL ROBOTS

The industrial robots are mainly used in the automation of the production processes and storage processes etc. Some of the applications are listed as follows:

- Assembly of the different parts in industries where there is repetitive work usually.
- Packing operations and handling heavy weight objects to pick and place them in different places required accordingly.
- Welding and Painting of heavy bodies which cannot be done by human always.
- Printing, measurement systems, etc.

2.4 BACKGROUND OF CNC

The numeric control for the machines tools were introduced in the early 1950's by Professor John T. Parsons. The numeric control machine was first built in the Massachusetts Institute of Technology (MIT) in the year 1953. It is built by the joint efforts of the US Air force, the MIT and the Parson's Co-operation.

Numerical control can be defined as a technique of controlling a machine or a tool by the direct loading of numeric data at some point of the system. The functions that can be controlled on the machine tools are displacement of sliding joints, spindle speeds, moving objects, tool selection etc. At first the displacements were only done by the concept of point-to-point, but later the linear and circular interpolation movements were introduced and developed which brought a significant growth in the field of these machines.

Numerical control is now employed in almost all the sectors of production, automation and storage. The development of the Computer Numeric Control (CNC) machines has led to the great advantage of the manufacturing industries. It has made easy the production process of each industry with simple and flexible ways to handle them.



Today the evolution of the CNC technology continues with the aim of increasing productivity, accuracy, speed, and facilitates the use of machine tools. The use of the numeric control favorably affects the cost of production by reducing the number of machines required in the industry. The CNC machines also maintain and improve the quality and performance of the manufacturing in the industry. At the same time, it also allows to achieve the operations with high accuracy which is impossible with the previously used conventional methods. In recent times the concept of the CNC machines has brought an enormous boom in the 3D printing field.

Applications of CNC:

- Milling machines
- Laser Cutting
- Drill press
- Lathe
- 3D printing, etc.

Advantages of CNC:

- High accuracy in manufacturing.
- Short production time.
- Reduced human error.
- Great manufacturing flexibility
- 2-6 axes of machining.

Dis-Advantages of CNC:

- Costly setup.
- Maintenance is difficult.
- Computers and programming knowledge required
- Skilled operators are required for running these machines.



2.5 EVOLUTION OF CNC

As the years passed the development of the numerical control machines has improved a lot. The main requirement being that the machines should be able to exchange information with all the factory using open standards. This will require the development of hardware, software and services that connect the machines to the supply chain, information systems and management systems.

In recent years, there have been major technological advances in the design of machine tools. Lately there has also been a development in the new software solutions based on automation and control technology for PC, offering CNC functionality with which there is improvement in the speed and precision of machining of very short segments.



Escuela Técnica Superior de Ingeniería del Diseño



CHAPTER 3

DESIGN OF CNC SYSTEM



3. DESIGN OF CNC SYSTEM

3.1 CONSTRUCTION OF THE SYSTEM

The CNC machine we are using in the project is constructed with the available hardware and mechanical material which is available in the laboratory. The machine is well constructed and also made use for the practices of the course work in the master.

The whole setup of the machine is made on a flat hard surface which is called the base of the system. A vertical framework of strong metal is made erected on the opposite sides of the base. This will be for supporting of the horizontal axis of the machine. A horizontal axis mounted of the same metal on the support of vertical framework. A conveyor belt is run along the axis with the help of wheels at the two ends of the horizontal axis. A DC motor is setup at one end of the axis for the running of the belt. The tool is mounted with the help of two cylindrical rods which help for the free movement along the horizontal axis. The tool will be moving in the x-direction along the cylindrical bars. Now the same setup is made up in the vertical direction for which the base is the horizontal axis. That means the conveyor belt setup in the y-direction is kept supported on the horizontal frame already mounted. The y-axis also has a conveyor belt, a DC motor mounted for itself. So, now the whole setup moves in their respective axes.

3.2 DC MOTORS

Electrical motors are those which we see daily everywhere around us in the world. Almost every electro-mechanical movement which we happen to see are caused by the AC or the DC motors. The motors which we used for our project are DC motors so we discuss about them here. The DC motors have a major importance in the present day industries in all the processes.

In simple words a DC motor can be defined as a device that converts direct current (electrical energy) into mechanical energy. The principle on which the DC motor works is, when a current carrying conductor is placed in a magnetic field, it will experience a torque and has a tendency to move. When there is an interaction between a magnetic field and an electric field they produce a mechanical force, and based on that the working principle of the DC motor is established. The tendency of the motor to produce the movement is known as the motoring action. If the direction of the current is reversed in the wire, then the direction of the motor movement also reverses.

The direction of rotation of the motor is given by a rule known as Fleming's Left hand rule. The rule states that if the index finger, middle finger and the thumb of your left hand are extended mutually perpendicular to each other and if the index finger represents the direction of magnetic field, the middle finger indicates the direction of current, then the thumb indicates the direction in which the force is experienced by the shaft of the DC motor.

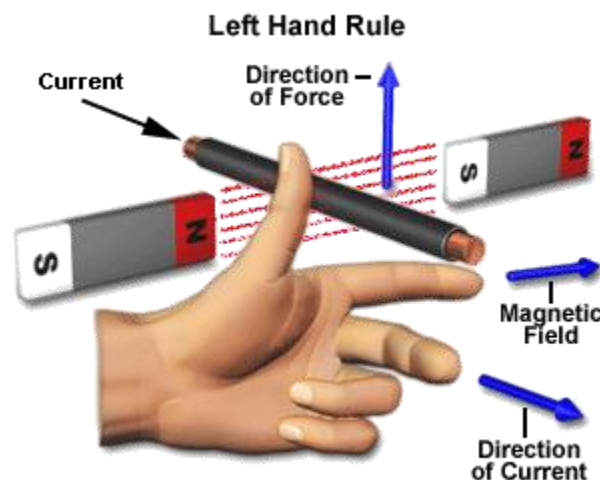


Figure 3. Fleming's Left Hand Rule

- The block diagram of a DC motor can be given as follows:

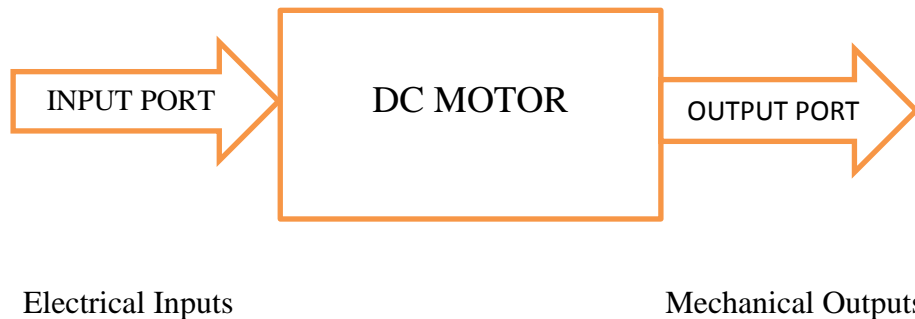


Figure 4. Block diagram of DC motor

There are different types of DC motors which are in use in the present industrial life. We have to consider several factors for choosing the DC motor for the required applications. Let us have a look at types of DC motors. This is what they are classified as.

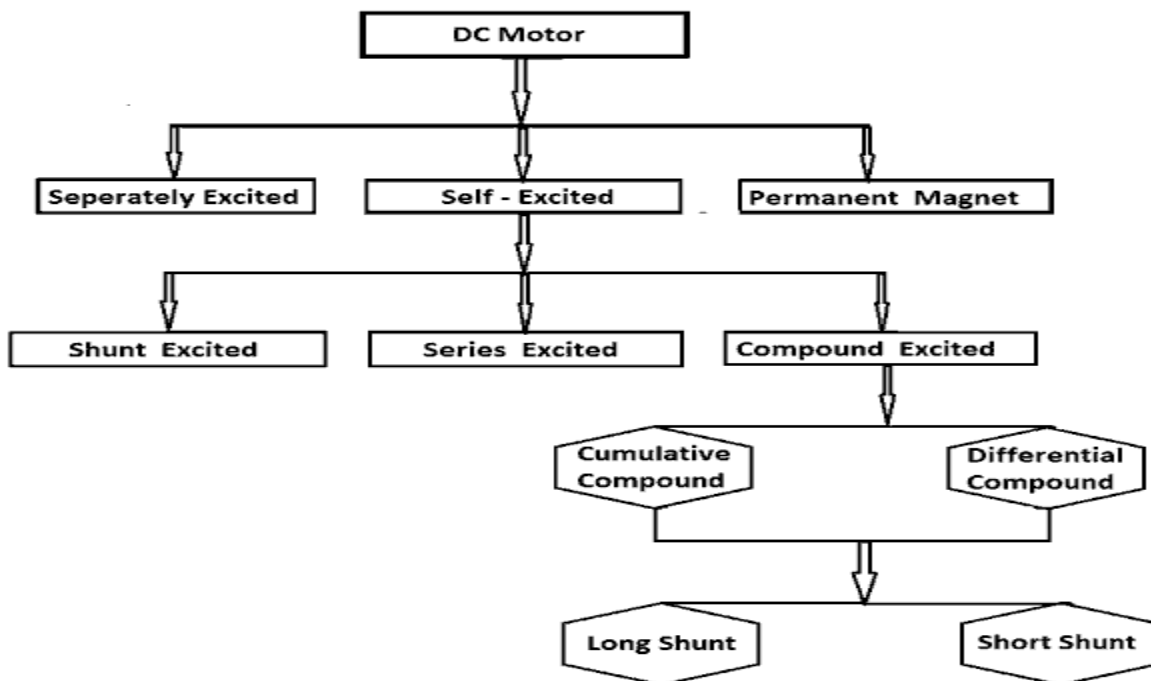


Figure 5. Types of DC motors



Separately Excited DC Motor:

As the name itself indicates in this type of the DC motor the supply is given separately to the field and the armature winding. The main reason that makes the difference in this type of motors is that, the armature current does not flow through the field windings, as it is energized from a separate external source of DC current.

Permanent Magnet DC motors:

This type of DC motors consists of an armature winding just as in a usual motor but it is not necessarily that it has field windings. The construction of these type of motors is in such a way that, radially magnetized permanent magnets are mounted on the inner layer of the stator core to produce the field flux. On the other hand, the rotor has a conventional DC armature with commutator segments and brushes.

Self-Excited DC Motors:

In this type of DC motors, the field winding is connected either in series or in parallel or partly in series and partly in parallel to the armature winding. Depending on this basis they are furtherly classified as:

➤ Shunt Wound DC Motors

In this case of Shunt wound DC motors, the field windings are exposed to the entire terminal voltage as they are connected in parallel to the armature winding. This is probably the most widely used motor in the industrial applications.

➤ Series Wound DC Motors

In case of the Series wound DC motors, the entire armature current flows through the field winding as it is connected in series to the armature winding. This type of motors



has a highest torque per current ratio of all the DC motors. They are mostly used as starting motors of cars and also in the elevators.

➤ **Compound Wound DC Motors**

The compound excitation characteristic in a DC motor can be obtained by the combination of both the series wound and shunt wound DC motors. This type of motors essentially contains the field winding connected both in series and also in parallel to the armature winding. The series field provides better starting torque and the shunt field provides better speed regulation.

• **Advantages of DC motor**

- ✓ Speed control over a wide range both above and below the rated speed.
- ✓ High starting torque.
- ✓ Accurate speed less speed with constant torque.
- ✓ Quick start, stop, reverse and acceleration movements.
- ✓ Easy to understand design.

• **Dis-advantages of DC Motor**

- ✓ High initial cost.
- ✓ High operational and maintenance cost due to presence of commutators and brush gear.
- ✓ Cannot operate in explosive and hazardous environment.

➤ **Applications of DC Motors**

- ✓ Steel mills, paper mills.
- ✓ Cranes and elevators.
- ✓ Electric trains, cars, etc.

The main characteristics we are looking for in the DC motor required for our application are as follows:

- Simple electronic control and easy to mount.
- Accuracy and repeatability in the movements.
- Good torque.
- Optimum electrical consumption.

The motor which we chose for our required application is Kelvin K30 37.62. This motors are manufactured by a Spanish company named KELVIN. They have been in the industry since 1927. The model specified has a gearbox which is used as the driver for the motor. In the name of the motor K30 is the model number of the gearbox and 37.62 is the series of the motor which is attached to the gearbox. So our required DC motor is the combination of gearbox and motor together which is Kelvin K30 37.62. Here is a view of the motor.

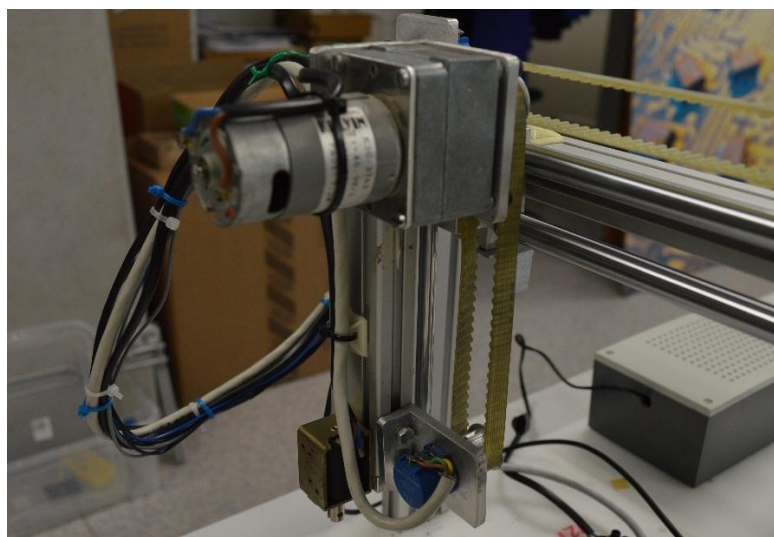


Figure 6. Kelvin K30 37.62 DC Motor

3.3 DRIVER CIRCUIT OF MOTOR

The electronic component that is necessary for the correct movement of the engines is called a "driver" and is specific for each type of engine. The driver is responsible for receiving the control signals from the microcontroller of the machine and send the order to the engines in the form of electrical signal with suitable shape and power, in addition to providing voltage and current suitable for the proper functioning of the engines.

The driver which we are using in our project is the L298N IC chip which is manufactured by ST Electronics. This chip is capable of controlling our motors which we are using in our project. It is a high voltage high current dual full bridge driver designed to drive DC as well as stepper motor too. The bridge circuit are usually used to control the speed and direction of the motors. It is available in two packages Multiwatt15 and PowerSO20. Here we use a Multiwatt15 package of L298N. The chip and its circuit diagram are as follows:

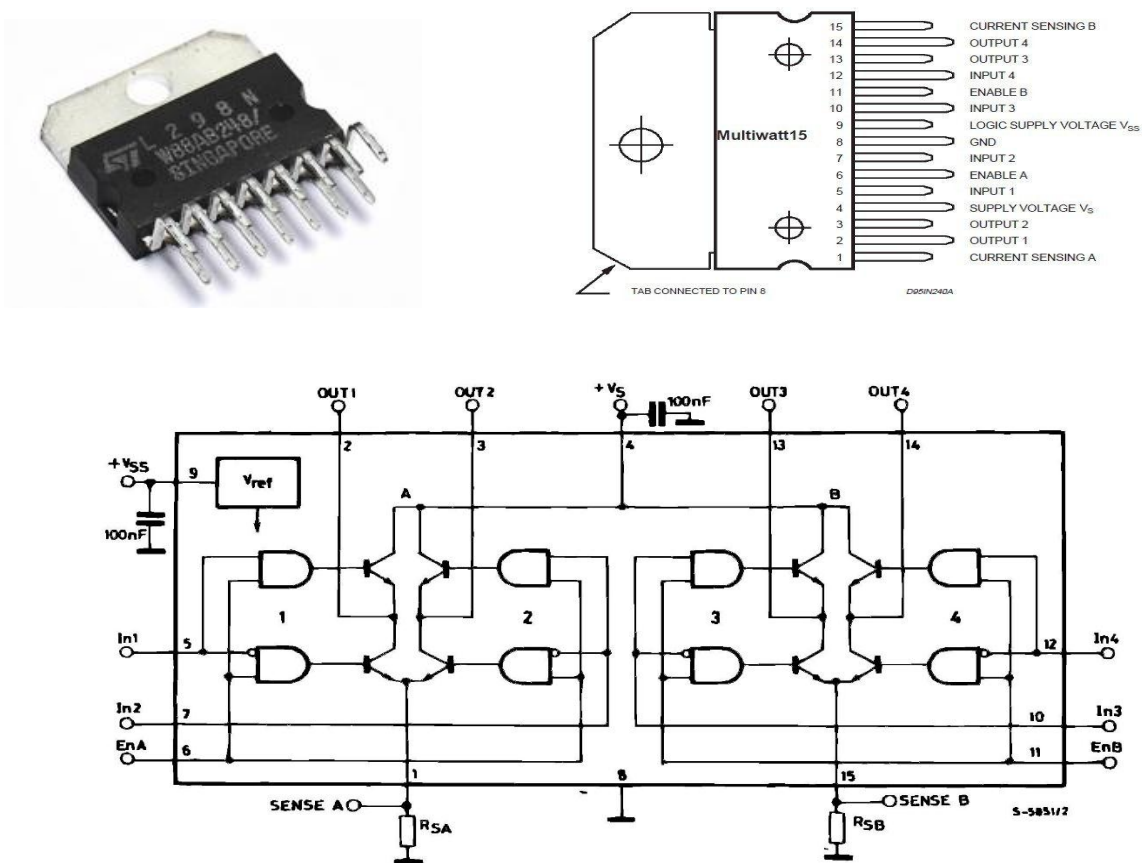


Figure 7. L298N Driver of ST Electronics

The L298N driver has the ability to run the motors with up to a voltage of 46V and with a current of 4A which is enough to control the motors. These values along with the characteristics of the motors are taken into account while configuring the software of the microcontroller. This is necessary to obtain a good torque and proper functioning of the system.

3.4 LIMIT SWITCHES

The limit switches are defined as those which prevent the travel of an object in a mechanism past some predetermined point, which are mechanically operated by the motion of the object itself. These are used for the controlling of the machines as part of the control systems as safety interlocks. A limit switch is an electromechanical device that has an actuator that is mechanically linked to a set of contacts. When a moving machine body comes into contact with the actuator the device operates to make or break an electrical connection.

Limit switches have a very ease of installation and are very reliable hence are used in a variety of applications in robotics and industries. They were first used to define the limit of travel of an object and hence the name came as “Limit Switches”. The switches we used here is as follows:



Figure 8. Limit Switch

The switch has three connections Normally Open (NO), Normally Closed (NC) and a ground. The micro-controller inputs of the sensors of the limit switches are very sensitive and less resistive to the disturbances. Hence they send back a return signal sometimes with poor contacts which results in improper functioning of the program. To avoid such situation, the limit switch circuit is integrated with a circuit of IC SN74L. This IC has 4 NAND gates which are useful in preventing the signal bouncing. This IC is connected to the microcontroller and it is capable of controlling two limit switches with each IC. The working of a Limit switch is shown in the below figure clearly.

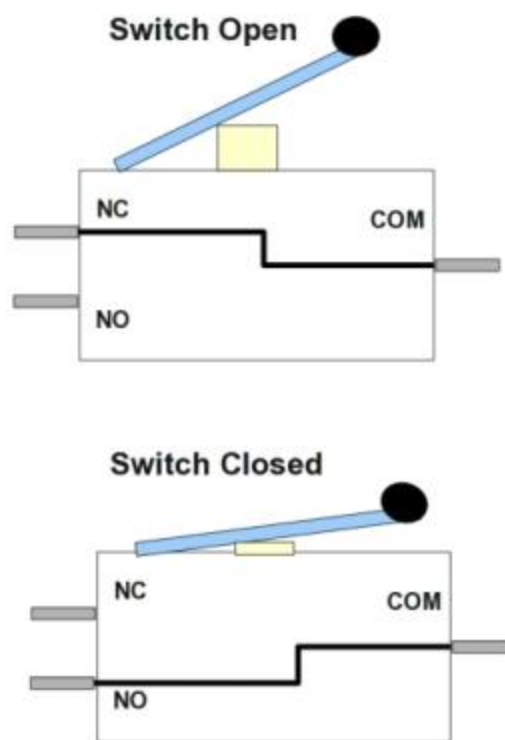


Figure 9. Working of a Limit Switch

According to the figure, when the switch is open, the internal connection is Normally closed (NC) so the motor keeps running and when the switch is closed the internal connection shifts to Normally Open (NO) and the motor is stopped. But in our project we are in a rectilinear motion in both directions we use the switch to just turn the motor in opposite direction whenever the switch is pressed or closed at each end.

3.5 MICROCONTROLLER

The micro-controller is considered as the heart of the machine. All the commands from the PC are passed to the machine through the microcontroller. It controls the whole mechanism of the machine. The decision to choose an appropriate microcontroller is made based on different schematic analysis and many other factors are considered and also which is easy to use for the future purposes.

The characteristics to be considered while choosing a microcontroller are:

- Inputs and Outputs
- Performance
- Control type (Processes)
- Memory size, technology, and expandability
- Software: Instruction set and programming modules
- Peripherals
- Physical and environmental behavior

Considering all the above stated characteristics we have chosen our board as STM32 F3 Discovery board with STM32 F303VCT6 microcontroller. We have studied this microcontroller in our master degree classes. We will see an in depth details of this microcontroller in the next chapter.

3.6 SOFTWARE

The software we use here for the programming is the KEIL software. This has ARM specific software package and is dedicated for programming the microcontrollers. The language used in the programming is Embedded C language.



Figure 10. Schematic of the system

3.7 CIRCUIT CONNECTIONS

The circuit connections of the motors and the microcontroller are made as follows:

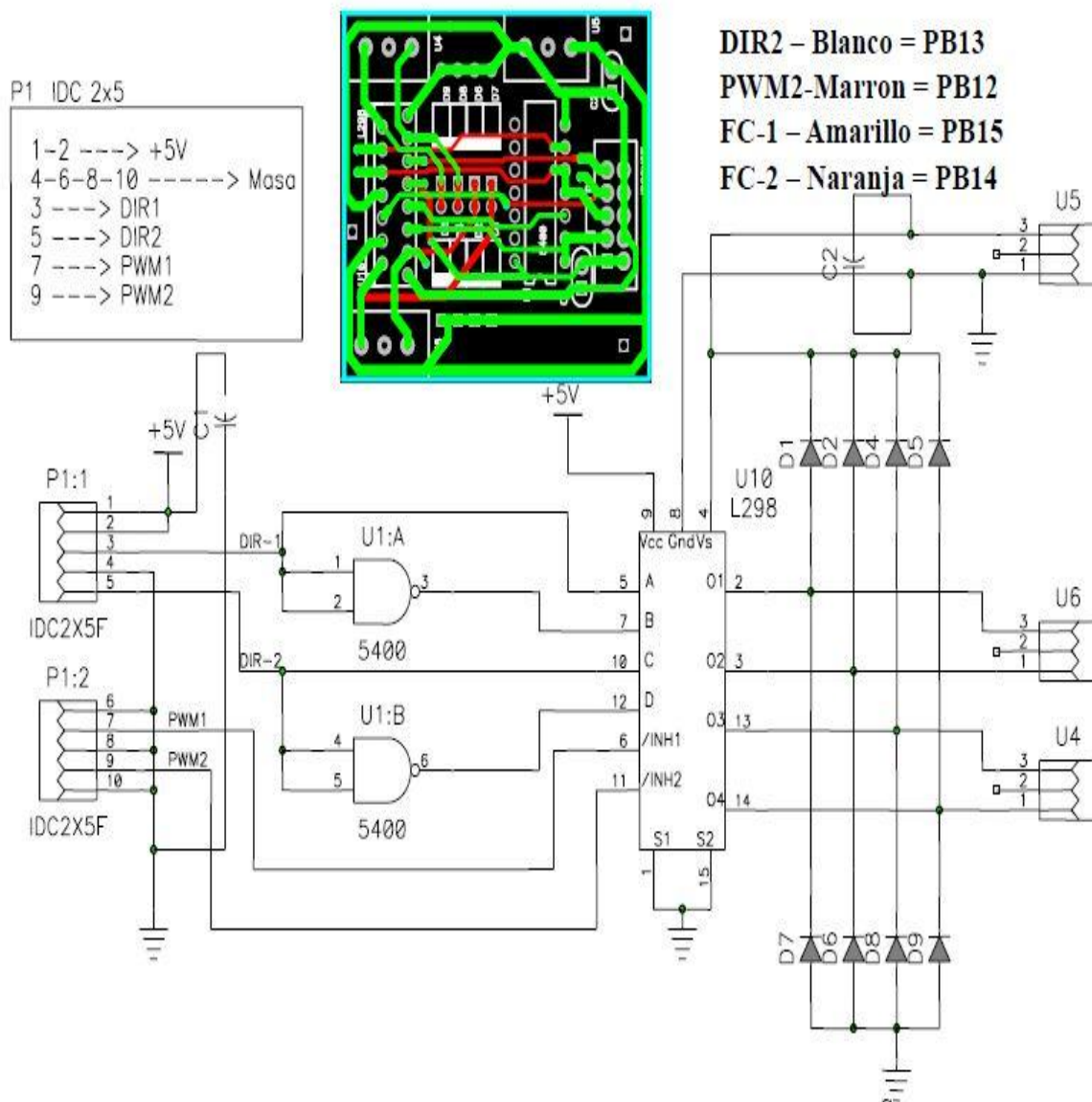


Figure 11. Circuit of DC motors connection to microcontroller

NOTE: This single IC board is capable of controlling two DC motors (X and Y axis motors here).

The circuit connections of the limit switches with the microcontroller are made as follows:

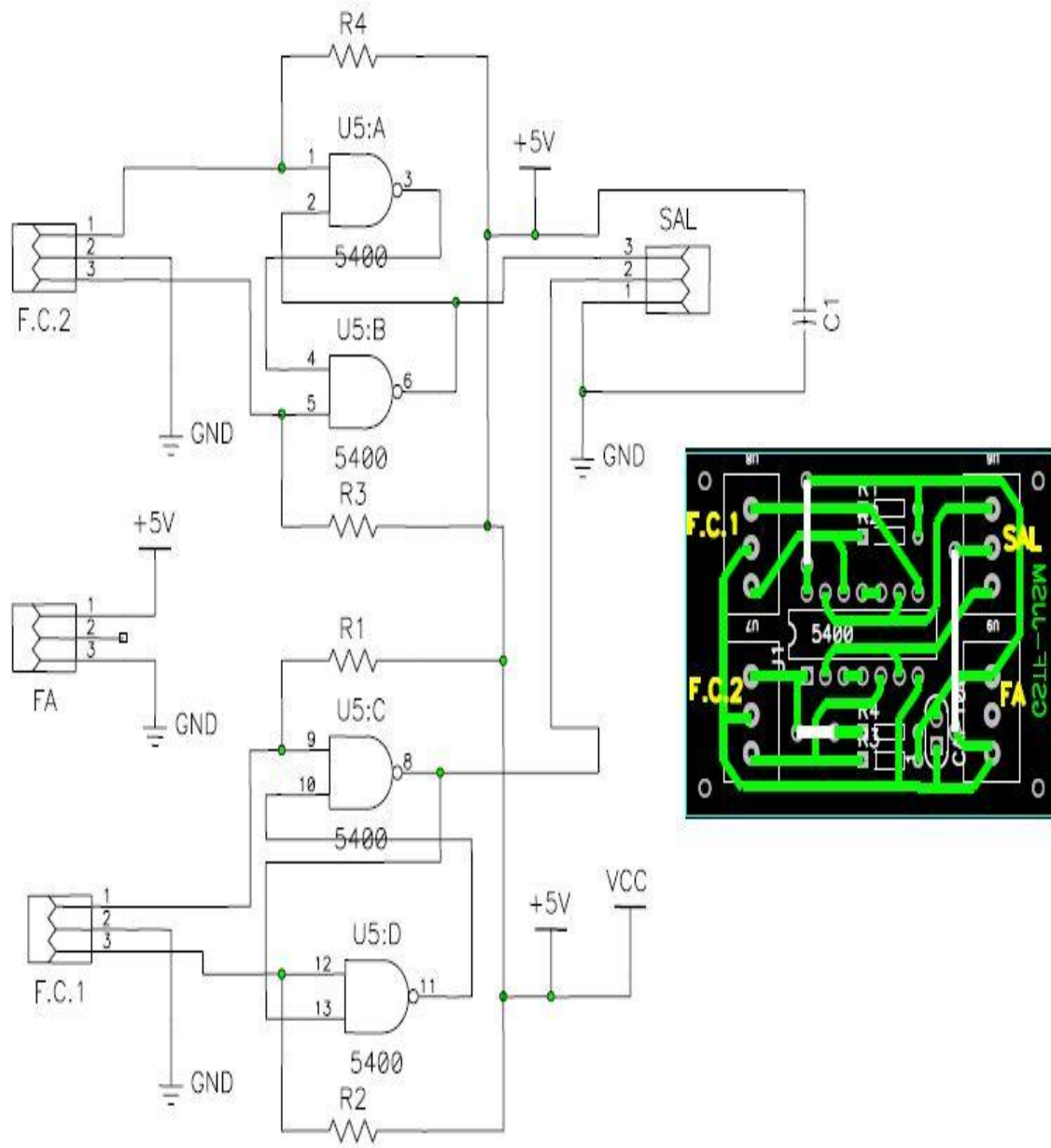


Figure 12. Circuit of limit switches connections to microcontroller

NOTE: The single IC board is capable of controlling two limit switches. We have two for each X and Y axes respectively.

Our machine after all the mechanical arrangements and the electrical connections given will look like this:

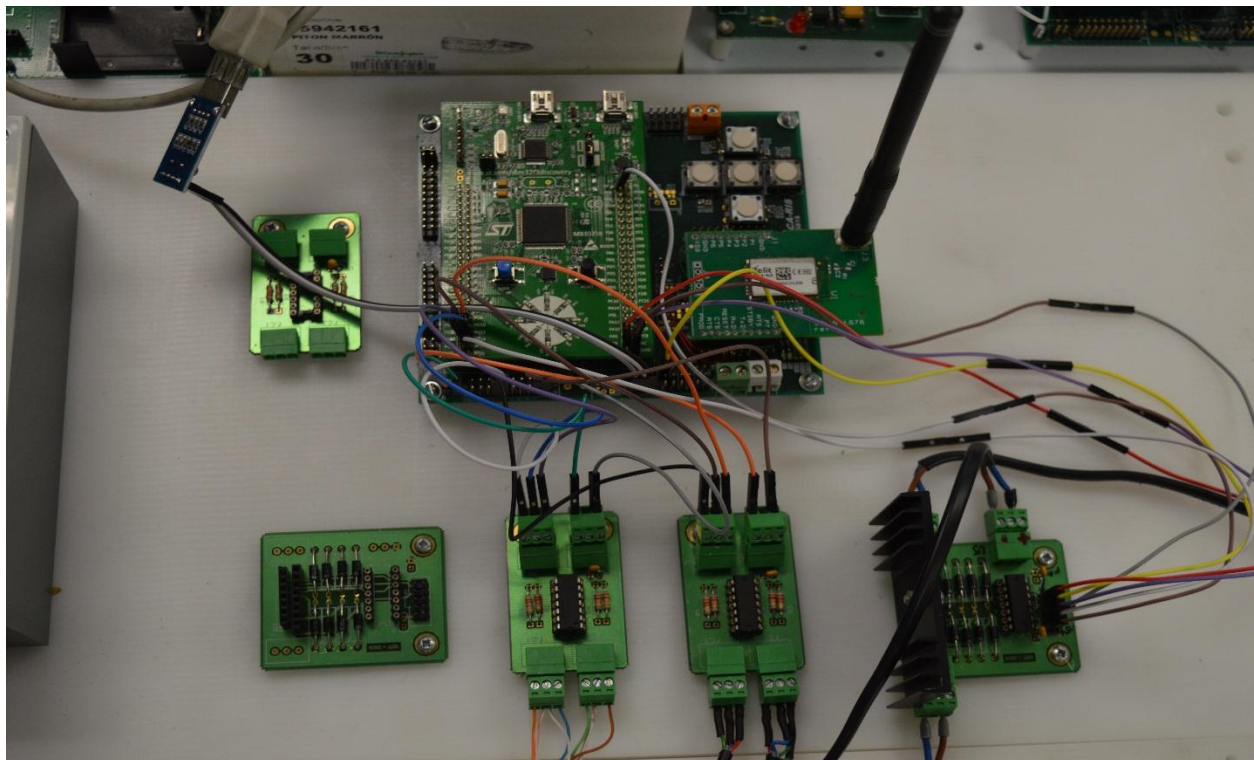
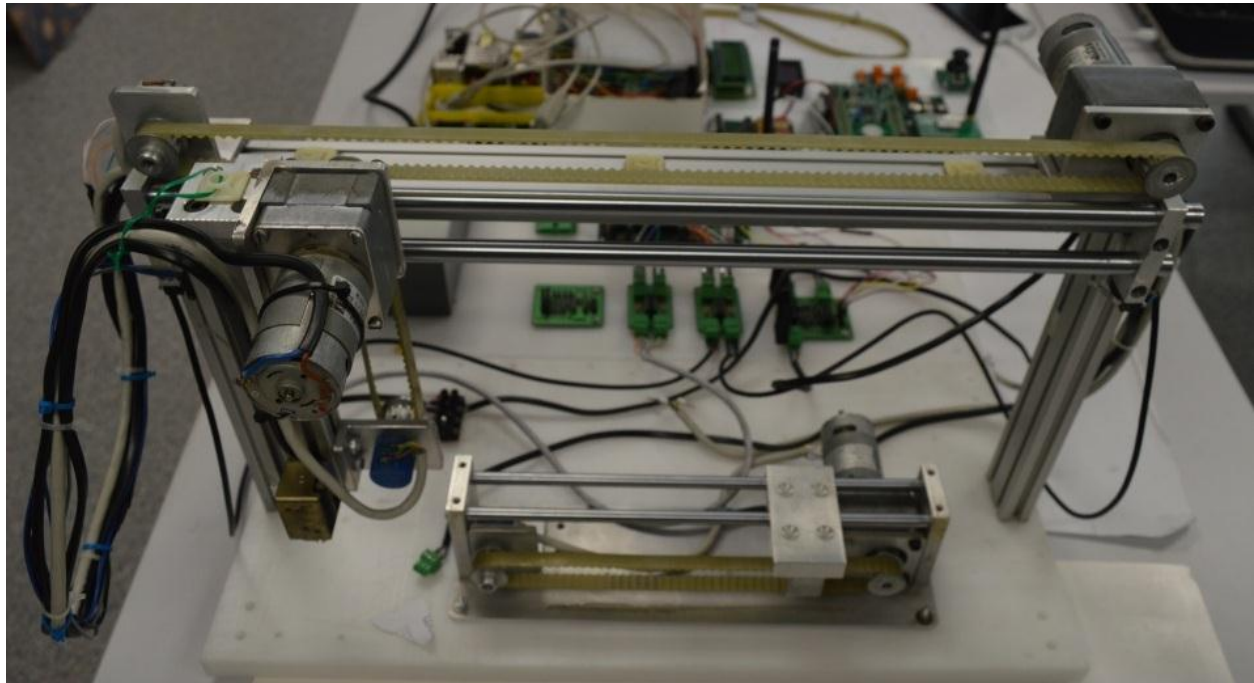


Figure 13. The project machine after all the connections



Escuela Técnica Superior de Ingeniería del Diseño



CHAPTER 4

STM32 F303VCT6 MICROCONTROLLER



4. STM32 F303VCT6 MICROCONTROLLER

4.1 INTRODUCTION TO MICROCONTROLLERS

According to Reese, Bruce and Jones, a microcontroller is a compact microcomputer on a single integrated circuit containing a core processor, memory and programmable input and output peripherals. Before the invention of microcontrollers, electronics were built using combinatory logic circuits which were implemented using a large number of chips. As a result, the size of the electronic devices is used to be much larger which is not so preferred in many cases and also it used to be very expensive. Due to this, some manufacturers thought of how they can build the same electronics using a more convenient and better way while eliminating the use of many chips. This led to the development of the microprocessors in the 1980's. This led to the reduction of production cost and also the cost of the products. The earliest microprocessors were 8 bit which included the Intel 8080 among others. These led to advancement and improved functionality of electronics since these microprocessors could do more compared to the logic circuits.

Later it was discovered that microprocessors could be easily updated. Despite the fact that microprocessors were a huge improvement, they did not fully bring down the cost and complexity of product design, hence still there is need for further improvement. For a microprocessor to do something useful we have to provide additional chips for input and output and also other support chips such as real-time clock chips and address decoders are needed. Chip manufacturing techniques which could integrate all the circuitry in one chip were invented. This was done with the use of improved silicon processing. Additional functions and features which were previously provided with a separate chip were included in the microprocessor chip. All of them put together in a single IC was called microcontroller. Microcontrollers are used for embedded applications. They are used in automatically controlled products and devices. The majority of them are used today in machinery such as automobiles, telephones, appliances, etc.

4.2 STM32 F3 DISCOVERY BOARD

The STM32 F3-series is the microcontrollers based on the ARM Cortex-M4 core and the most recent series manufactured from ST Microelectronics. The ARM architecture is a 32-bit Reduced Instruction Set Computer processor which has a good power saving attribute. Due to this, the ARM central processing units are widely being used in mobile phones. Our microcontroller we are using here is ARM Cortex M4 core based STM32 F303VCT6. The schematic of the microcontroller is as follows:

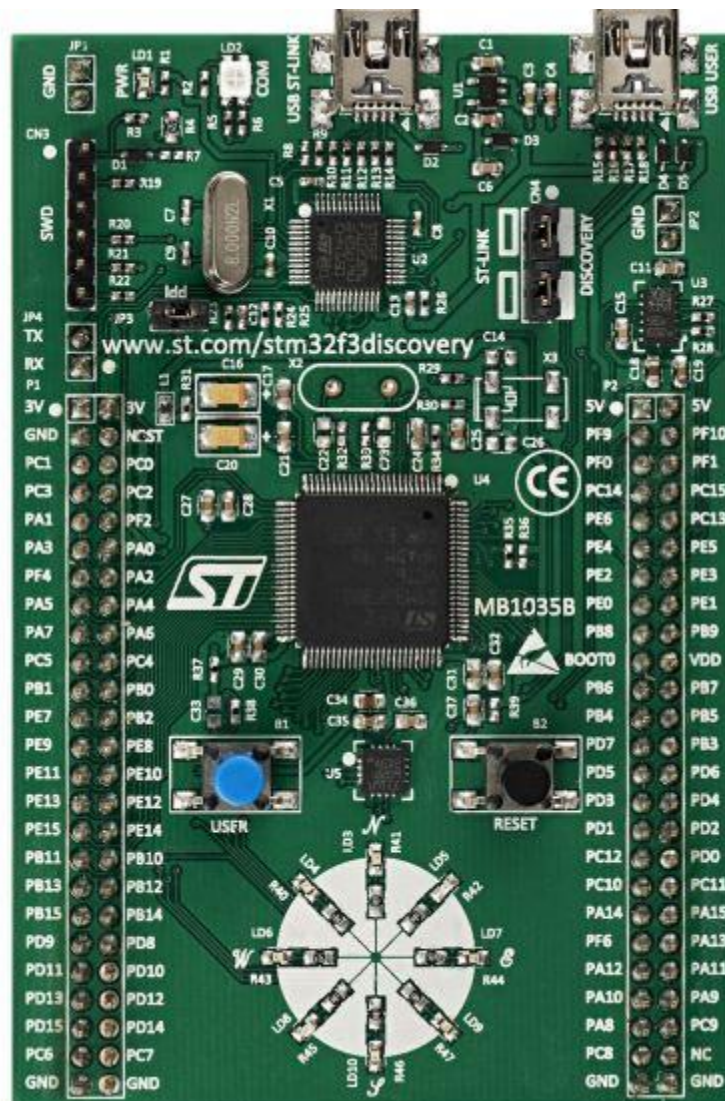


Figure 14. Schematic of STM32 F3 Discovery Board



The STM32 F3 Discovery board has the following features:

- STM32 F303VCT6 microcontroller featuring 256KB of flash memory and 48KB of RAM in LQFP100 package.
- On board ST-Link/V2 or ST-Link/V2-B
- USB user with Mini-B connector
- Board power supply from a USB bus or from a 3V or 5V external supply voltage.
- USB ST-Link functions
 - Debug port
 - Virtual COM port with ST-Link/V2-B only
 - Mass storage with ST-Link/V2-B only
- Ten LED's
- Two Push buttons (User and Reset)
- Extension header for LQFP100 I/O's for quick connection to prototyping board and easy probing.

STM32 F3 series microcontrollers have found a wide range of applications since they have led to a reduction of cost at the application level and simplicity in design. This is as a result of their Ultra-fast comparators, Op-amp with programmable gain among other features. The main features of this device include high performance and architecture. This boosts execution to control algorithms, better code efficiency, and data code reliability among others. The device also has outstanding power efficiency which leads to flexibility in reducing power consumption for applications requiring advanced analog peripherals and low power modes. Another attractive feature is that it has superior and innovative peripherals with up to 18 times resolution power more than previous series with a clock timer oscillating up to 144MHz.

Microcontrollers find their applications in alternative energy generation, automotive and transportation, consumer and portable electronics, industries, medicine and smart grid. In day-to-day life, microcontrollers are used in fire detection and safety devices, temperature sensing and control devices. They are also employed in process control devices which monitor ongoing production processes. Metering and measuring instruments also use microcontrollers.



4.3 STM32 CUBEMX

STM Cube is an ST Microelectronics original initiative to ease developers life by reducing the developers efforts, time and cost. STM32 Cube covers STM32 portfolio. It includes the STM32CubeMX which is a graphical software configuration tool that allows generating C initialization code using graphical wizard. This software helps to configure all the ports, timers, inputs and outputs, channels and all on a single platform before the coding can be done. This is a very user friendly software tool which eases the work of the developer. The main features of the software are:

- Intuitive STM32 microcontroller selection.
- Microcontroller graphical configuration.
 - Pinout with automatic conflict resolution.
 - Clock tree with dynamic validation of configuration.
 - Peripherals and middleware functional modes and initialization with dynamic validation of parameter constraints.
 - Power sequence with estimates of consumption results.
- C code project generation covering STM32 Microcontroller initialization compliant with IAR, KEIL and GCC compilers (Here Keil).
- Generation of configuration reports.
- Power consumption calculation.
- Easy switching to another microcontroller belonging to the same series.

STM32CubeMX is a graphical tool that allows configuring STM32 microcontrollers very easily and generating the corresponding initialization C code through a step-by-step process.

Step one consists of selecting the STMicroelectronics STM32 microcontroller that matches the required set of peripherals. Our microcontroller is STM32 F303VCT6.

Then we must configure each required embedded software thanks to a pinout-conflict solver, a clock-tree setting helper, a power-consumption calculator, and a utility performing MCU peripheral configuration (GPIO, USART, ...) and middleware stacks (USB, TCP/IP, ...).

Finally, we must launch the generation of the initialization of C code based on the selected configuration. This code is ready to be used within several development environments. The user code is kept at the next code generation. .pdf and .csv reports can also be generated to document user configuration work. C code generation covers only peripheral and middleware initialization. It is based on STM32Cube HAL firmware libraries. Our configured microcontroller through STM32CubeMX looks as follows:

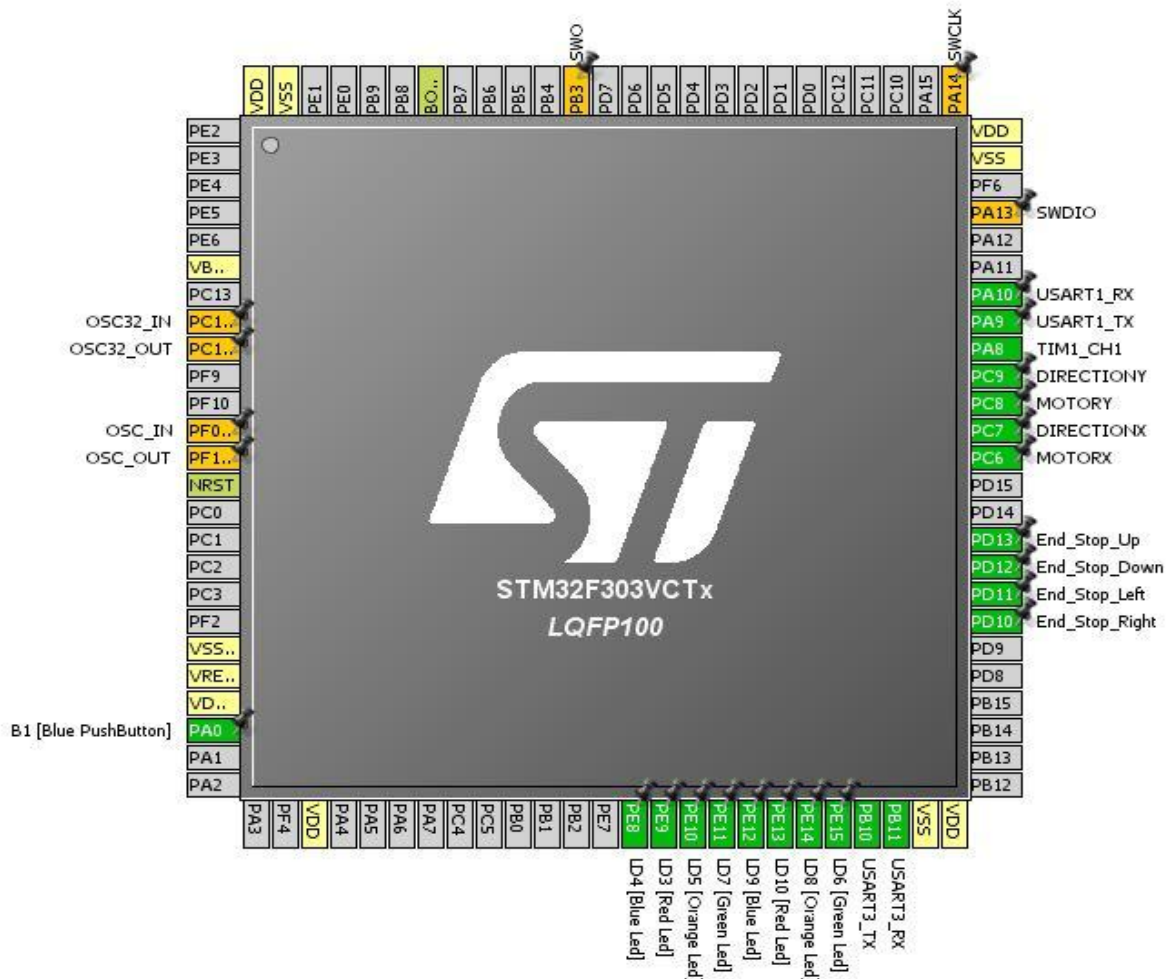


Figure 15. STM32CubeMX main window after configuring the microcontroller.



Escuela Técnica Superior de Ingeniería del Diseño



CHAPTER 5

RADIO FREQUENCY COMMUNICATION MODULE



5. RF COMMUNICATION MODULE

5.1 INTRODUCTION

Communications is of two types, wireless and wired. Wireless communication is defined as the transfer of the data or power between two or points that are not connected by any means of electrical conductors. This eases the design complexity of any process or machinery. The most commonly used wireless technology uses the radio frequency waves. With radio waves we can communicate between very short distances and also between many million kilometers' distances also. This type of electromagnetic (radio) wave frequencies lies in the range of 3 KHZ to 300 GHZ. It has a wide range of applications in cellular telephones, GPS units, garage door openers, wireless computer mouse and keyboard, headsets, satellite television, etc. LTE, LTE-Advanced, Wi-Fi, and Bluetooth are the common modern wireless technologies used in the present generation. The main advantage of wireless communications is that they allow the services such as long range communications that are impractical and impossible to implement with the help of wires.

5.2 RF MODULE

A radio frequency module is defined as an electronic device which is used to transmit and/or receive the radio signals between devices. In embedded systems the most desired way to communicate is in the wireless mode. The wireless communications can be done through optical or radio frequency (RF) communications. Here we use the RF communications. Good electronic radio design is normally complex to implement because of the sensitivity of the radio circuits and the accuracy of the components and layouts required to achieve operation on a specific frequency. In addition to this, the module should be carefully monitored to ensure the performance is not affected during the manufacturing process. RF module typically incorporates a transmit or receive circuit, printed circuit board, antenna and a serial interface for communication to the host processor. The RF module we use in our project is Telit LE50-868.

5.3 TELIT LE50-868

Telit LE50-868 is one of the short range wireless RF module which is designed by the TELIT company. The Telit LE50 family is comprised of multi-band, multi-channel radio modules operating on an advanced proprietary embedded stack that is easy to integrate and use in point-to-point or star network topologies. The modules are easy to integrate, with small form factor and low price. They have very less power consumption too. They operate in 433MHz and 868 MHz bands and feature low power standby mode, efficient wake up on radio and link budget of 123 dB (119 dB for 433 MHz EU-only).

This LE modules operates in the 868 MHz ISM license free frequency band with transmission power up to 25mW. It organizes in a Star Network topology over a Telit proprietary protocol ideal for use in replacing communication over cables such as RS485 links (Profibus, Modbus) and half-duplex RS232 links (transparent mode), with wireless technology. With the new smart repeater function, it can also be used to create communication networks in a star topology and for long-chain communications. The main features of this module are as follows:

- Excellent Receiver sensitivity.
- ‘AT’ mode or Hayes mode for configuration.
- Listen Before Talk (LBT) for collision avoidance.
- Certified RF modules, LGA xE form factor, TTL RS232 interface.
- Download over the air (DOTA).
- Repeater mode (bridge function).
- AES128 data encryption available.
- Cyclic wake up: wakes up periodically and listens to radio link.
- Industrial temperature range.

The RF module which we are using in the project looks like this:



Figure 16. Telit LE50-868 RF Module with its support board.

Some of the other important product features of Telit LE50-868 RF Module are as follows:

Product Features

- Range up to 2000m
- Flash up to 128 kB, RAM of 4 kB, EEPROM of 2 kB
- 4 timers, 32.768 kHz real time clock (RTC)
- Configurable output power
- 9 I/O ports maximum available

Electrical and Sensitivity

- Output power up to 14 dBm
- Power supply: 2.0 to 3.6 V
- Board Consumption at 25 mW
 - Rx: 26 mA
 - Tx: 45 mA
- Sensitivity (CER < 10-3): -109 dBm

- Stand-by consumption:
 - External wake-up (interrupt): $1\mu\text{A}$
 - Cyclic wake-up (internal timer running): $3\mu\text{A}$

Data

- Serial Data rate: Up to 115.2 kbps.
- Radio Data rate: from 4.8 kbps to 115.2 kbps.

5.4 CONFIGURATION OF TELIT LE50-868

We have the RF Module now for our project and the next thing we need to do is to set the parameters for our application purpose. For this we use a tool called S-One management tool which is developed to Telit Company itself for its users to make it easy. So we take the module along with supporting microcontroller board and connect it to the laptop via USB connection. Then we start the application and go on as follows:

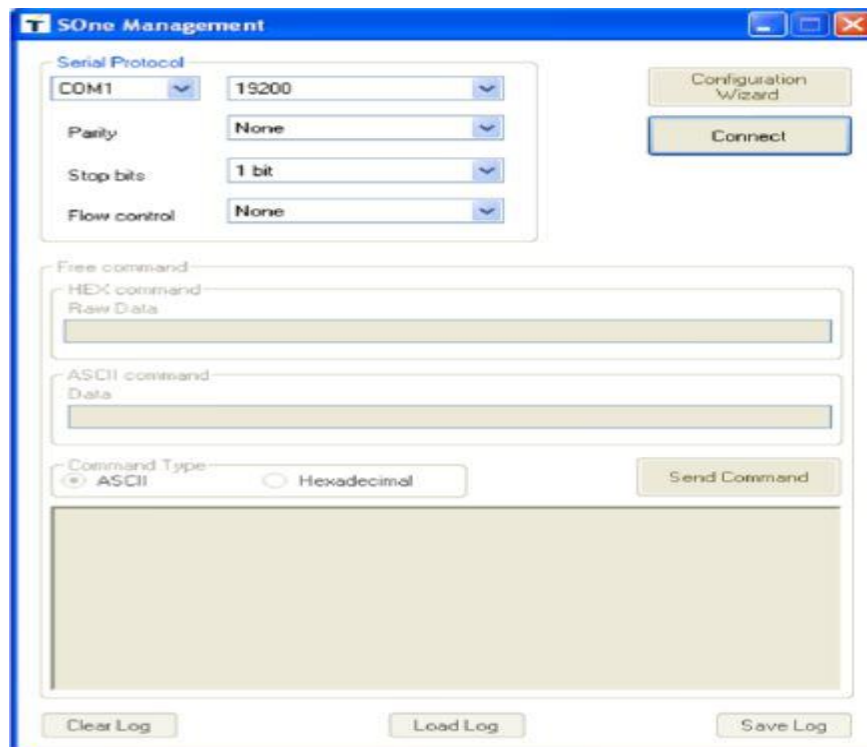


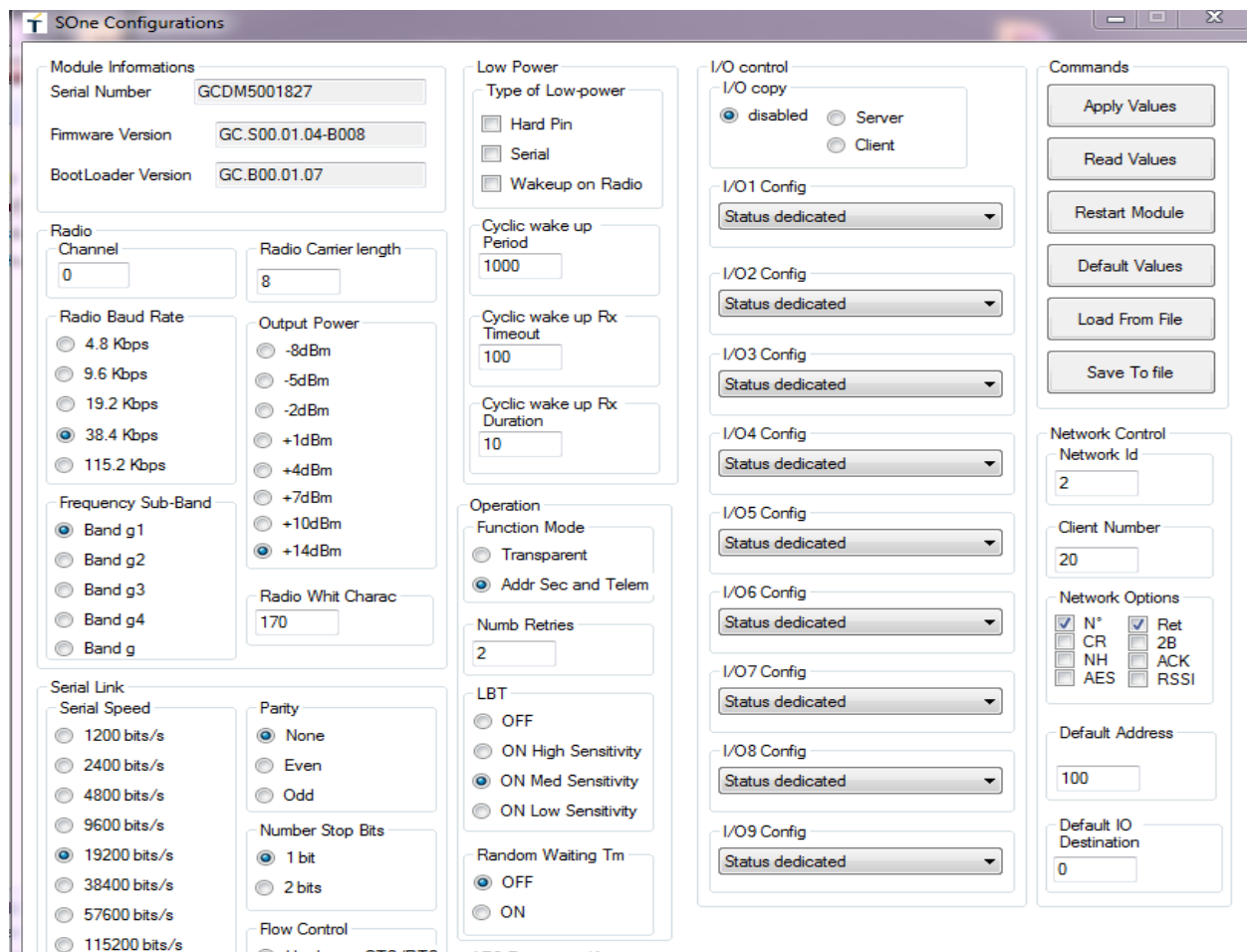
Figure 17. S-One management tool opening window

As shown above in the figure we need to select the options shown for us regarding the serial port, baud rate, parity, stop bits and flow rate. Then we click on ‘Connect’ button to get connected. The values for us are:

Serial Protocol: COM1 Baud rate: 19200 bits/sec (default). Parity: none

Stop bits: 1 Flow control: None.

Once we get connected there is a small box at the bottom of same window which shows IN and OUT values for us to check whether we have connected properly or not. When we are done correctly then click on configuration wizard option to configure the remaining parameters and registers of the module. The next window which comes is as follows:



The screenshot shows the 'S-One Configurations' window with the following sections and settings:

- Module Informations:**
 - Serial Number: GCDM5001827
 - Firmware Version: GC.S00.01.04-B008
 - BootLoader Version: GC.B00.01.07
- Radio:**
 - Channel: 0
 - Radio Carrier length: 8
 - Radio Baud Rate: 38.4 Kbps (selected)
 - Output Power: +14dBm (selected)
 - Frequency Sub-Band: Band g1 (selected)
 - Radio Whit Charac: 170
- Serial Link:**
 - Serial Speed: 19200 bits/s (selected)
 - Parity: None (selected)
 - Number Stop Bits: 1 bit (selected)
 - Flow Control: Hardware: CTS/RTS (selected)
- Low Power:**
 - Type of Low-power: Hard Pin (selected)
 - Cyclic wake up Period: 1000
 - Cyclic wake up Rx Timeout: 100
 - Cyclic wake up Rx Duration: 10
- Operation:**
 - Function Mode: Addr Sec and Telem (selected)
 - Numb Retries: 2
 - LBT: ON Med Sensitivity (selected)
 - Random Waiting Tm: OFF (selected)
- I/O control:**
 - I/O copy: disabled (selected)
 - I/O1 Config: Status dedicated
 - I/O2 Config: Status dedicated
 - I/O3 Config: Status dedicated
 - I/O4 Config: Status dedicated
 - I/O5 Config: Status dedicated
 - I/O6 Config: Status dedicated
 - I/O7 Config: Status dedicated
 - I/O8 Config: Status dedicated
 - I/O9 Config: Status dedicated
- Commands:**
 - Apply Values
 - Read Values
 - Restart Module
 - Default Values
 - Load From File
 - Save To file
- Network Control:**
 - Network Id: 2
 - Client Number: 20
 - Network Options:
 - N* (checked), CR (checked), NH (checked), AES (checked)
 - Ret (checked), 2B (checked), ACK (checked), RSSI (checked)
 - Default Address: 100
 - Default IO Destination: 0

Figure 18. The Configuration Wizard window of S-One management tool.



When the configuration wizard window is opened as shown above the values that it displays are the default values that are stored in it. When we click the button 'Read values', then it reads the values of the module that is connected and displays for us. Now we need to change them with the required values for us. We are using two RF modules here one for the transmission end and another for the receiver end. The values which are shown above in the figure are of the values of the transmission end. For the receiver end module also almost all the values are same except for a couple of values called Client number and default address are to be changed. After setting all the values click on the 'Apply Values' button to configure all the parameters and the registers of the module. At the end a window pops out showing 'Registers setting completed' and then you are done. Our values are as follows:

For transmission module:

- Network ID = 2
- Client Number = 1
- Default Address = 100

For Receiver module:

- Network ID = 2
- Client Number = 100
- Default Address = 0

After the configuration of the module is done we assemble it on the microcontroller we have with the help of a support board. Next we have to write a program for the module using the Keil and Embedded C and then pass it through the microcontroller for its working. We use the USART ports of the microcontroller for this purpose. We will be seeing the programming part in the coming chapter in detail.

5.5 ANALOG JOYSTICK

For the control of our machine in the wireless mode we use an analog joystick as the controller. It is a very simple joystick which is almost similar to the one we use in the gaming controller joysticks. It is a very simple and small one as shown in the figure below:



Figure 19. Analog Joystick



This analog joystick gives us more accurate control than the simple directional joysticks with buttons or mechanical switches. It has two potentiometers built in its construction which acts as sensors and provide us with a variable voltage depending on rotation of the joystick head around its shaft. So with this joystick we can be able to control anything in two directions.

As shown in the figure it has five connections namely Vcc, Gnd, X, Y and Key. As we know Vcc and Gnd means supply and ground. X and Y are the two analog inputs from which the analog data is taken. And Key is a digital input which can be used as a selection button or push button or a switch. If we do not need a selection pin, then we can use only four pins.

Here the two analog inputs we are using for our motors in two axes. We read the analog data from joystick and convert into digital data using the convertors of the microcontroller and then transmit it to the other board which is responsible of receiving the data and controlling the machine.



Escuela Técnica Superior de Ingeniería del Diseño



CHAPTER 6

SOFTWARE PROGRAMMING



6. SOFTWARE PROGRAMMING

6.1 KEIL SOFTWARE

The software we used for the programming, compiling and debugging in our project is the Keil Software. It is a software development toolchain. The version which we are using here is Keil Microcontroller Development Kit (MDK) Version 5 which is a latest version released with complete software development environment for a wide range of ARM Cortex-M based microcontroller devices. It includes the μ vision IDE/Debugger, ARM C/C++ Compiler, and middleware components which are easy to learn and use.

We need to assign and configure all the input/output pins, timers, channels, GPIO ports, USART channels with the help of the STM32CubeMX tool as we have discussed earlier. We name all of the entries with the names which are convenient to us for the usage.

We have two motors to control in two axes for our machine, so we require two GPIO output ports from the MCU to control the movements of the motor and two GPIO output ports to control their directions. We totally have four limit switches, two for each axis and their signals need to be given to MCU through the GPIO input ports, so we use four input ports for them. We are using a joystick whose two analog inputs are given to the microcontroller which needed to be converted to digital so we use two ADC convertors. Each ADC convertor has different channels where we use one channel from each convertor. And we use two Usart channels which are responsible for the data transmission and reception between the two microcontroller boards we are using for the control of our machine. Each Usart channel requires two ports one for transmission and other for the reception. So in total we use four ports for the Usart purposes. Other than these ports there are other internal clock ports, ground and power supply ports which can be used as per our requirements. All these ports are named as per our convenience and configured accordingly with the help of the STM32CubeMX tool. A list of all the ports that we have used is also generated in a file and the table is as shown below:



PIN NUMBER LQFP100	PIN NAME (Function After Reset)	PIN TYPE	ALTERNATE FUNCTION(S)	LABEL
6	VBAT	Power		
8	PC14-OSC32_IN*	I/O	RCC_OSC32_IN	OSC32_IN
9	PC15-OSC32_OUT*	I/O	RCC_OSC32_OUT	OSC32_OUT
12	PF0-OSC_IN*	I/O	RCC_OSC_IN	OSC_IN
13	PF1-OSC_OUT*	I/O	RCC_OSC_OUT	OSC_OUT
14	NRST	RESET		
20	VSSA/VREF-	POWER		
21	VREF+	POWER		
22	VDDA	POWER		
23	PA0**	I/O	GPIO_INPUT	B1[BLUE PUSH BUTTON]
28	VDD	POWER		
29	PA4	I/O	ADC2_IN1	ADC2_IN1
36	PB1	I/O	ADC3_IN1	ADC3_IN1
39	PE8**	I/O	GPIO_OUTPUT	LD4[BLUE LED]
40	PE9**	I/O	GPIO_OUTPUT	LD3[RED LED]
41	PE10**	I/O	GPIO_OUTPUT	LD5[ORANGE LED]
42	PE11**	I/O	GPIO_OUTPUT	LD7[GREEN LED]
43	PE12**	I/O	GPIO_OUTPUT	LD9[BLUE LED]
44	PE13**	I/O	GPIO_OUTPUT	LD10[RED LED]
45	PE14**	I/O	GPIO_OUTPUT	LD8[ORANGE LED]
46	PE15**	I/O	GPIO_OUTPUT	LD6[GREEN LED]
47	PB10	I/O	USART3_TX	USART3_TX
48	PB11	I/O	USART3_RX	USART3_RX
49	VSS	POWER		
50	VDD	POWER		

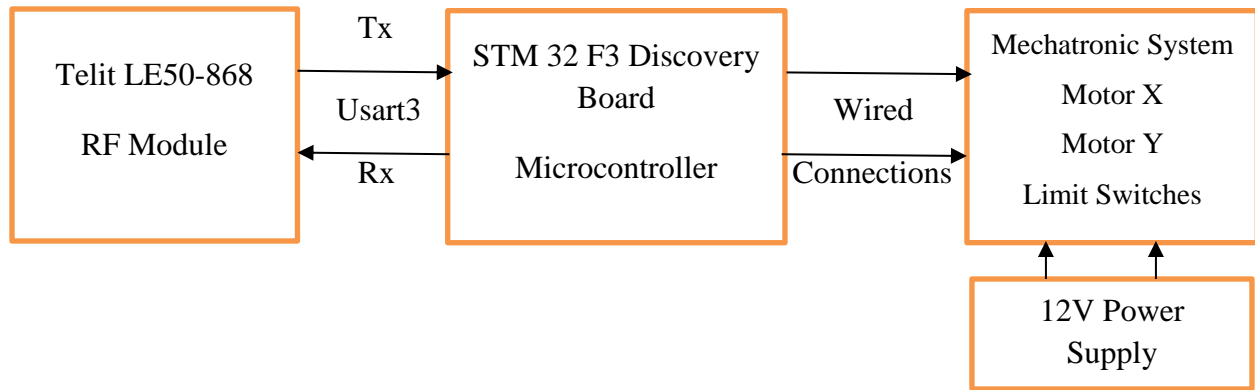
57	PD10**	I/O	GPIO_INPUT	END_STOP_RIGHT
58	PD11**	I/O	GPIO_INPUT	END_STOP_LEFT
59	PD12**	I/O	GPIO_INPUT	END_STOP_DOWN
60	PD13**	I/O	GPIO_INPUT	END_STOP_UP
63	PC6**	I/O	GPIO_OUTPUT	MOTORX
64	PC7**	I/O	GPIO_OUTPUT	DIRECTIONX
65	PC8**	I/O	GPIO_OUTPUT	MOTORY
66	PC9**	I/O	GPIO_OUTPUT	DIRECTIONY
68	PA9	I/O	USART1_TX	USART1_TX
69	PA10	I/O	USART1_RX	USART1_RX
72	PA13*	I/O	SYS_JTMS-SWDIO	SWDIO
74	VSS	POWER		
75	VDD	POWER		
76	PA14*	I/O	SYS_JTCK-SWCLK	SWCLK
89	PB3*	I/O	SYS_JTDO- TRACESWO	SWO
94	BOOT0	BOOT		
99	VSS	POWER		
100	VDD	POWER		

** The pin is affected with an I/O function.

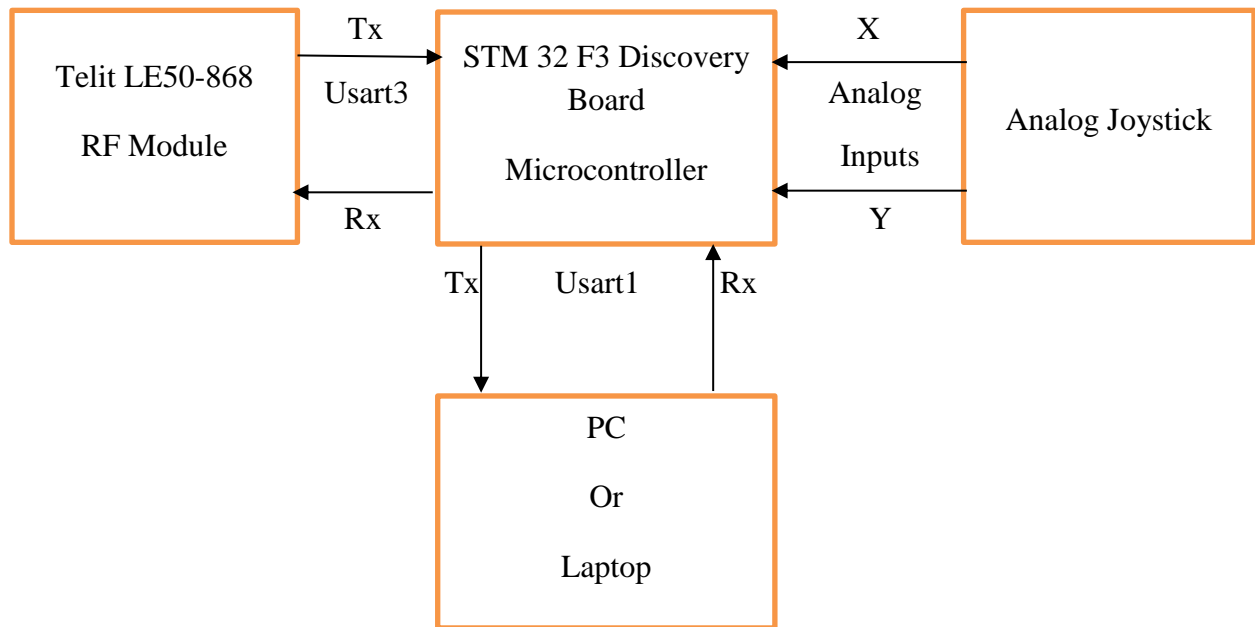
* The pin is affected with a peripheral function but no peripheral mode is activated.

6.2 BLOCK DIAGRAM

After all the ports are configured and ready then we start to implement the program code in Embedded C with the help of Keil software and the libraries of the microcontroller. A rough way of the program flow to complete the code and the block diagram of the whole setup are as follows:



Receiver End



Transmission End

Figure 20. Block Diagram of the Mechatronic system of the project

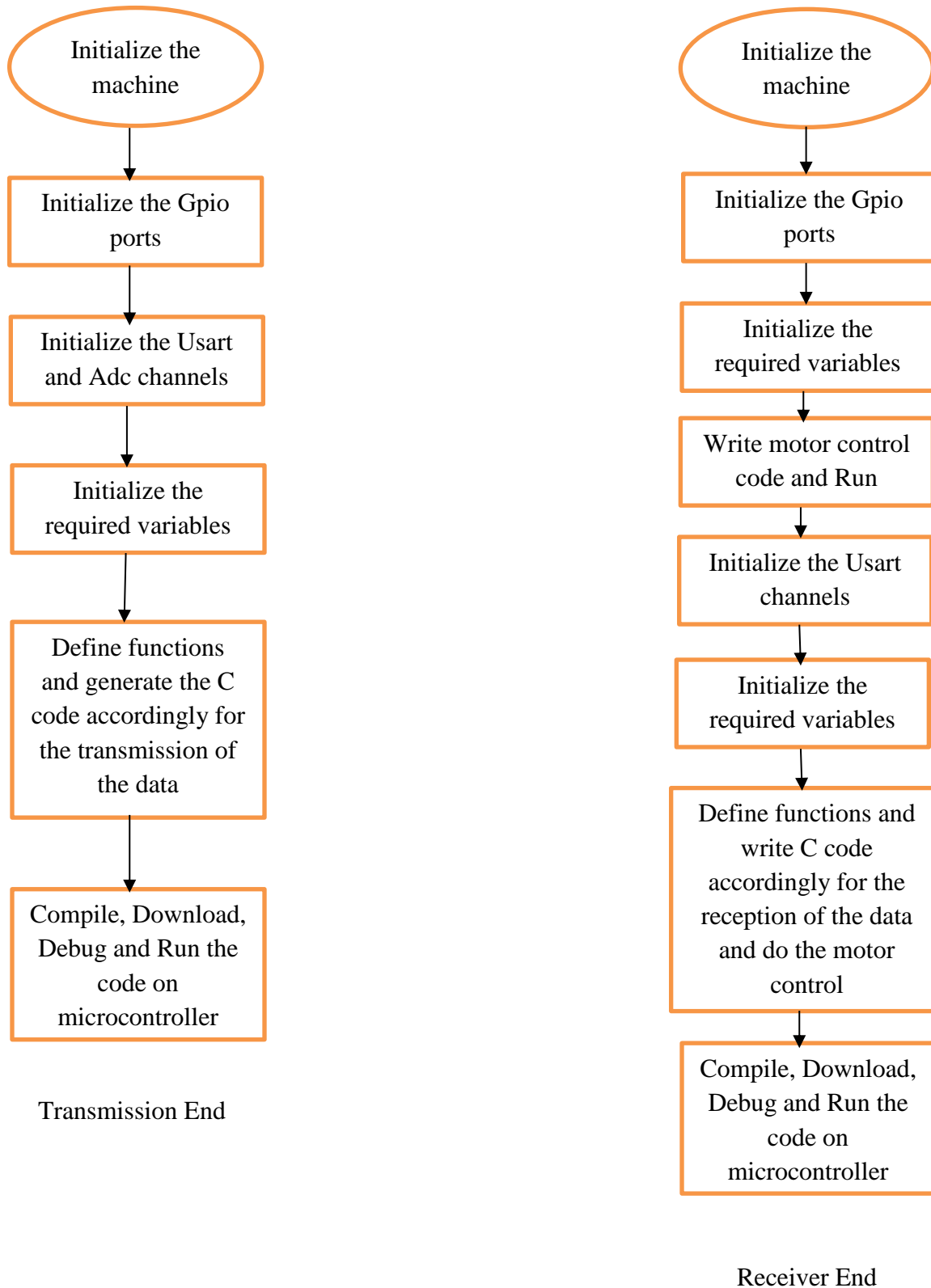


Figure 21. General flow chart of work flow



6.3 PROGRAM

The program is generally started by configuring the Inputs, Outputs, timers, adc, usart channels but we have already done that with another tool we discussed earlier STM32CubeMX, thanks for it makes our work easy. We can now directly declare some variables we need and define some functions which we use regularly at different places. The functions which we define can be called in at any point of time in the program.

We have two motors and we name them as motor_x and motor_y. We have defined many functions for the different position of the motors and all are almost similar as follows,

```
void motorx_fwd(void)
{
    HAL_Delay(100);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOC, DIRECTIONX_Pin, GPIO_PIN_RESET);
}
```

If we call this function, then the result is the motor_x moves in the forward direction. Similarly, for the backward, up, down, home, halt and stop positions are also defined. The limit switches also have their own way of functioning and we know their status by the set or reset state of their respective pins. The motor control part of the code is written using these functions in an infinite while loop using the if conditions.

In the next part we do the transmission part of the code. For that we are reading the analog inputs from the joystick. These analog inputs are to converted into digital for transmitting to the other board. This is done using the adc convertors which are in-built in our microcontroller but we need to define a function to use it. The adc function is something as follows:



```

void data_x(void)
{
    //uint16_t value_x;
    HAL_ADC_Start(&hadc2);
    if (HAL_ADC_PollForConversion(&hadc2, 10) != HAL_OK)
    {
    }
    value_x = HAL_ADC_GetValue(&hadc2);
    HAL_Delay(50);
    HAL_ADC_Stop(&hadc2);
}

```

By using this function, the analog input is converted into digital and ready for the transmission. We need to remember that all the adc, usart, timers which we use must be initialized in our main program.

Next we need to do the transmission of the data from one MCU to another MCU which is responsible for the control of the motors. For this we need to define some Usart functions for transmission and reception as well. Some functions we used here are as follows:

```

//Call back function
void HAL_USART_TxCpltCallback(USART_HandleTypeDef *UsartHandle)
{
    if (UsartHandle->Instance == USART3)
    {
        Usart3Ready = SET; //set the flag
    }
    else if (UsartHandle->Instance == USART1) //or &huart1
    {
        Usart1Ready = SET; //set the flag
    }
}

```

And there is a receiver call back function also as follows,

```
char Rx_idx, Rx_data[2], Rx_Buffer[100], Transfer_cplt;

void HAL_UART_RxCpltCallback(USART_HandleTypeDef *UartHandle)
{
    if (UartHandle->Instance == USART1) //or &huart3 ?
    {
        Uart3Ready = SET; //set the flag
    }

    else if (UartHandle->Instance == USART3)
    {
        uint8_t i;
        {
            if (Rx_idx==0)
            {
                for (i=0;i<100;i++)
                    Rx_Buffer[i]=0;
            } //clear Rx_Buffer before receiving new data

            if (Rx_data[0]!=13) //if received data different from ascii 13 (enter)
            {
                Rx_Buffer[Rx_idx++]=Rx_data[0]; //add data to Rx_Buffer
            }
            else //if received data = 13
            {
                Rx_idx=0;
                Transfer_cplt=1;//transfer complete, data is ready to read
            }

            HAL_UART_Receive_IT(&huart3,(uint8_t*) Rx_data, strlen(Rx_data)); //activate UART receive interrupt every time
        }
        Uart1Ready = SET; //set the flag
    }
}
```

Once the data is received at the receiver board it should get to know as to which command it has to follow to control the motors. For this purpose, we have used a separate function and called it in our main program which is as follows:



```
void rf_received_data(void)
{
    if(Rx_Buffer[Rx_indx] == 'A')
    {
        motory_up();
    }

    if(Rx_Buffer[Rx_indx] == 'B')
    {
        motorx_fwd();
    }

    if(Rx_Buffer[Rx_indx] == 'C')
    {
        motory_down();
    }

    if(Rx_Buffer[Rx_indx] == 'D')
    {
        motorx_rev();
    }
}
```

Now using all these functions, we finish the logic of the code in the main file with the help of calling these functions. After everything is done we compile the code for errors and fix if any. Then we debug the code again and then load the program in the microcontrollers of transmission end and reception end respectively. We should note that we write here two main codes each one for transmission and reception end MCU's respectively. We also use a hyper-terminal window in the PC to see and check the data that's being transmitted. That is the reason we are using two Usart channels here.



Escuela Técnica Superior de Ingeniería del Diseño



CHAPTER 7

RESULT AND CONCLUSION



7. RESULT AND CONCLUSION

Firstly, we check the code for the control of two motors in two axes in wired mode by passing our code directly to the motors through the microcontroller and the result of the movements of both axes is really satisfactory and accurate. Although the limit switches are a bit too sensitive and to be taken care as sometimes an environmental disturbance may change its value. But this can be avoided by having a very tiny delay to read the value again and re-bounce if its caused by disturbances.

Next we work in the wireless mode using the rf module. After good study of the software and hardware components and performing the code the resultant movements are satisfactory. The joystick functions well and the module works in a good range of distance. The motors run smoothly and the power supply of 12V is capable of driving them without any problem. A hyper terminal window is used on a laptop screen to check for the command that are being transmitted correctly or not.

By concluding we can say that the objectives that we had in mind are mostly attained. The study needed to build this project has been very handy in different aspects and also we can use it for more advancements.

7.1 FUTURE SCOPE

We have only done a two-dimensional machine here which can be further extend to 3D machine on the same basis. It may get a little tricky but it's definitely attainable. Also the Rf module can be replaces with a very high range wi-fi module so that we can be able to control the same machine by sitting in a different location with the help of an observation camera attached to it. This type of control is very helpful where we need to do some works in a hazardous



environment where a human can't work. At the end of the machine we can use any source as an effector like a pen for drawing, a robotic arm for pick and place of an object, a drill for making some artwork or a 3D printer and so on can be handled easily in the wireless mode just with a joystick or any controller. The control algorithms for the movement purpose can be improvised more to work in an efficient way.



Escuela Técnica Superior de Ingeniería del Diseño



CHAPTER 8

REFERENCES



8. REFERENCES

8.1 BOOKS

- Perryman, Cooper. 2012. *Microcontrollers*. Delhi: Research World. Wilmshurst, Tim. 2007.
- Reese, Robert B., J. W. Bruce, and B. A. Jones. 2009. *Microcontrollers: from assembly language to C using the PIC24 family*. Boston, MA: Course Technology.
- Van, S. T. (2001). *Programming microcontrollers in C*. Eagle Rock, Calif: LLH Technology Pub.
- Wilmshurst, T. (2007). *Designing embedded systems with PIC microcontrollers: Principles and applications*. Amsterdam: Newnes.

8.2 MANUALS

- User manual of Discovery kit with STM32F303VC MCU (ST Microelectronics).
- Getting started with software and firmware environments for the STM32F3DISCOVERY Kit.
- STM32CubeMX for STM32 configuration and initialization C code generation (ST Microelectronics).
- Telit RF Module - Star Network Protocol Stack User Guide.

8.3 WEB

- https://en.wikipedia.org/wiki/Numerical_control
- <http://www.slideshare.net/abhijityelam/cnc-introduction>
- <http://www.globalspec.com/reference/10789/179909/chapter-3-ac-and-dc-motors-dc-motors-dc-motor-types>
- <http://electrical-engineering-portal.com/4-types-of-dc-motors-and-their-characteristics>
- <http://aatechniek.nl/wp-content/uploads/Catalogo-KELVIN-2008.pdf>
- <http://www.st.com/en/evaluation-tools/stm32f3discovery.html>
- http://www.st.com/content/st_com/en/products/motor-drivers/brushed-dc-motor-drivers/l298.html
- <http://42bots.com/tutorials/arduino-joystick-module-example/>